DBMS Project

**Festival Management System**

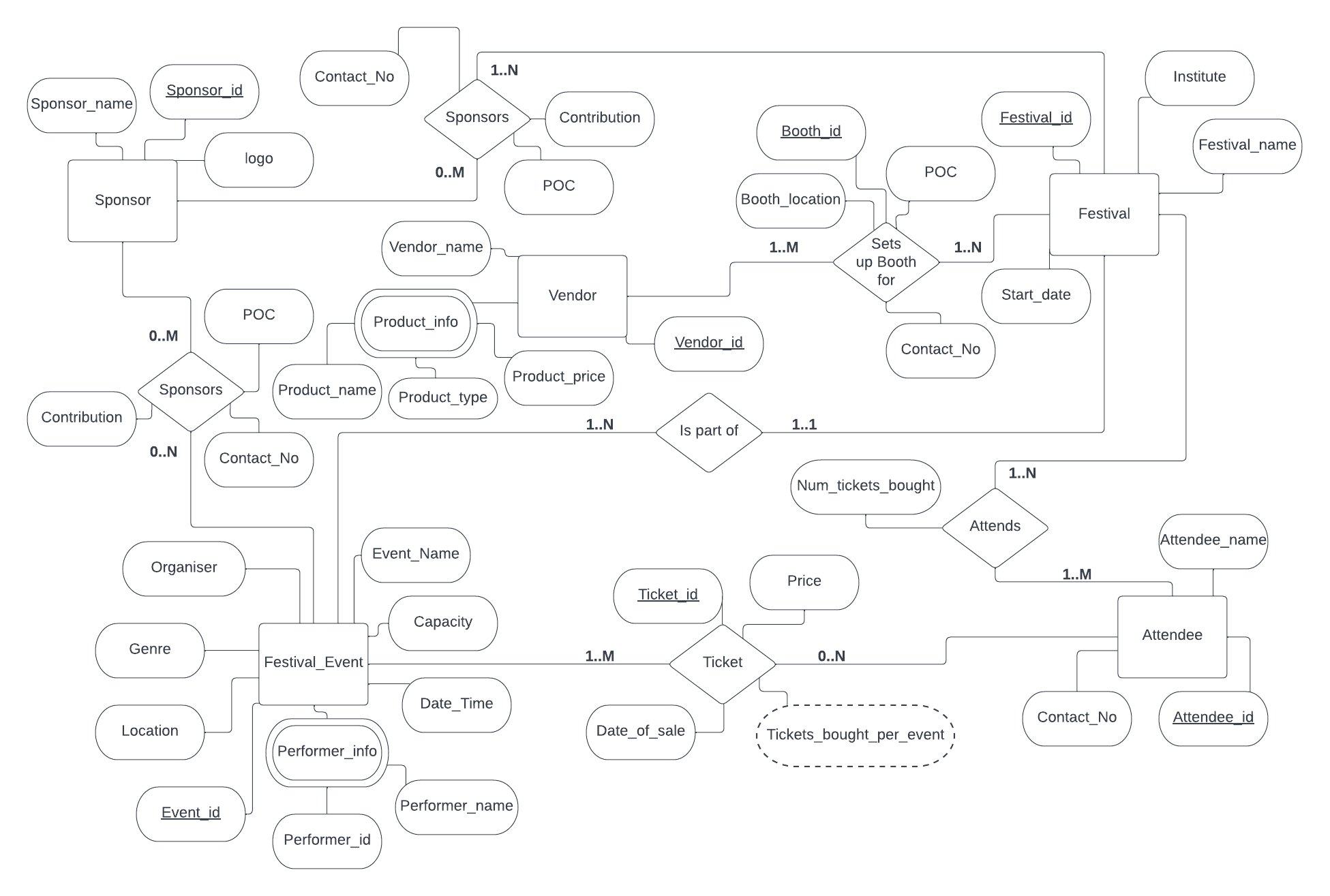
By-Vidit Benjwal-2021A7PS0004P

Aneesh Kabra-2021A7PS0442P

**INDEX**

1. **ER Diagram**
2. **Conversion to 1NF Relational Schema  
   + Functional Dependencies**
3. **Normalization to 2NF/3NF Relational Schema**
4. **Database**
5. **SQL Queries + Output**
6. **Drive Link**

**ER Diagram**



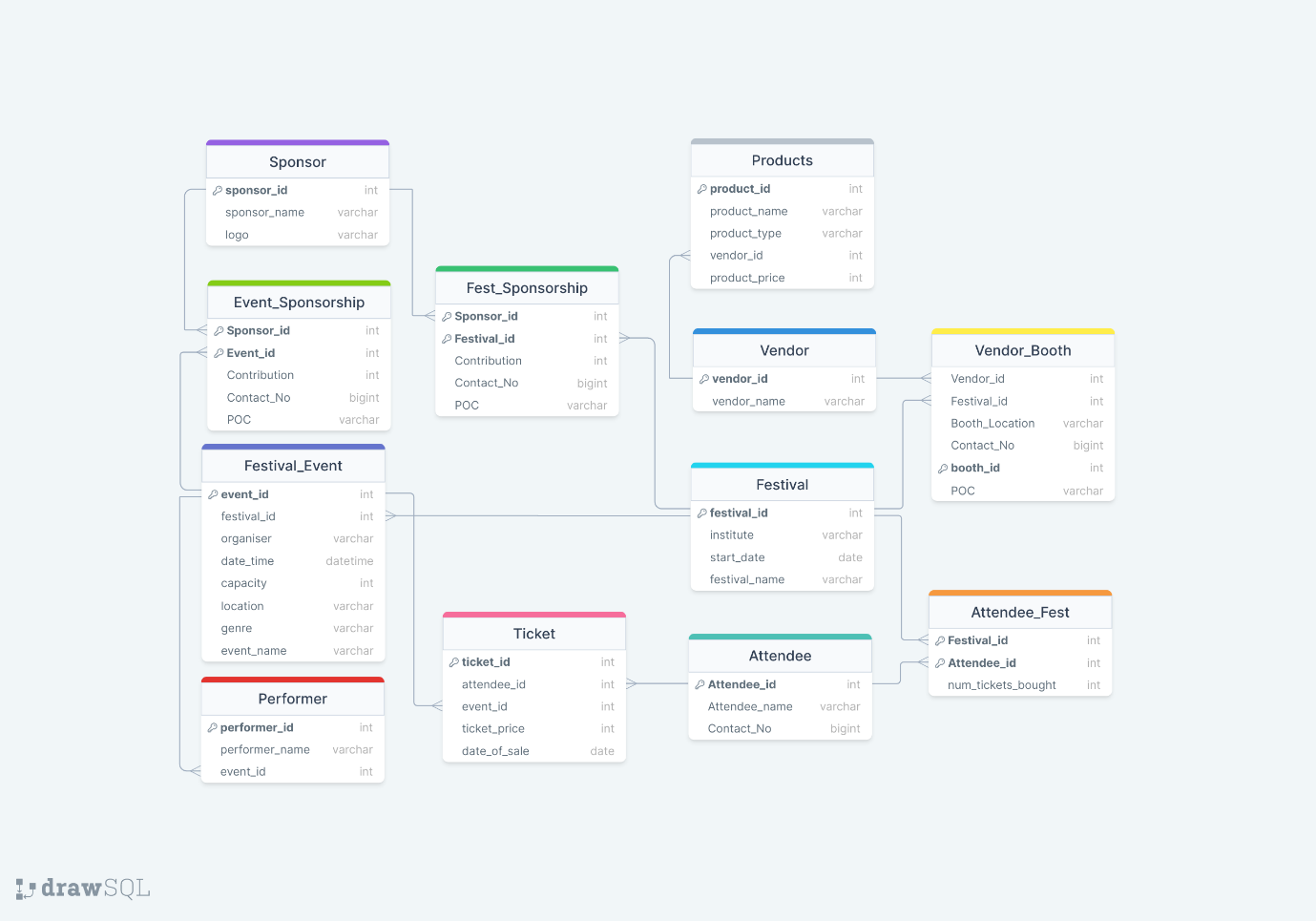
This ER diagram consists of 5 entities (**Sponsor, Festival, Attendee, Festival\_Event, Vendor**).

1. **Sponsor** - Has 3 simple attributes - Sponsor\_name, Sponsor\_id, logo. The primary key for this entity is *Sponsor\_id.*
2. **Festival** - Has 4 simple attributes - *Festival\_name, Festival\_id, Institute, Start\_date.* The primary key for this entity is *Festival\_id.*
3. **Attendee** - Has 3 simple attributes - *Attendee\_id, Attendee\_name, Contact\_No.* The primary key for this entity is *Attendee\_id*.
4. **Festival\_Event** - Has 7 simple attributes and 1 complex attribute which has 2 simple attributes under it - *Event\_id, Event\_Name, Capacity, Date\_Time, Location, Organiser, Location, Performer\_Info. Performer\_Info* is a complex attribute and has *Performer\_id, Performer\_name* under it. The primary key for this entity is *Event\_id.*
5. **Vendor** - Has 2 simple attributes and 1 complex attribute which has 3 simple attributes under it - *Vendor\_id, Vendor\_name, Product\_info. Product\_info* is a complex attribute and has *Product\_name, Product\_type, Product\_price* under it. The primary key for this entity is *Vendor\_id.*

These entities have a total of 6 relationships between each other which are -

1. **Sponsor** sponsors **Festival** -This is a many-to-many relationship as one festival can have many sponsors and one sponsor can sponsor many festivals. Sponsor shows partial participation while Festival shows total participation in this relationship. This is because a sponsor does not necessarily need to sponsor a festival but a festival must have at least one sponsor. The independent attributes of this relationship are 3 simple attributes - *Contact\_No, Contribution, POC*.
2. **Sponsor** sponsors **Festival\_Event** - This is a many-to-many relationship as one festival event can have many sponsors and one sponsor can sponsor many festival events. Both Sponsor and Festival\_Event show total participation in this relationship. This is because a sponsor does not necessarily need to sponsor a festival event and a festival event also need not necessarily have a sponsor. The independent attributes of this relationship are 3 simple attributes - *Contact\_No, Contribution, POC*.
3. **Festival\_Event** Ticket **Attendee** - This is a many-to-many relationship as one attendee can attend many events and one event can have many attendees.Attendee shows partial participation while Festival\_Event shows total participation in this relationship. This is because an attendee of a festival does not need to necessarily attend any event in it but a festival event has to have at least one attendee. The independent attributes of this relationship are 3 simple attributes - *Ticket\_id, Price, Date\_of\_sale* and 1 derived attribute - *Tickets\_bought\_per\_event*. The *Ticket\_id* attribute is the primary key for this relationship.
4. **Attendee** Attends **Festival** - This is a many-to-many relationship as one attendee can attend many festivals and one festival can have many attendees. Both Attendee and Festival show total participation in this relationship. This is because an attendee must attend at least one festival to qualify as an attendee and a festival must have at least one attendee for it. The independent attribute of this relationship is 1 simple attribute - *Num\_tickets\_bought*.
5. **Festival\_Event** is part of **Festival** - This is a many-to-one relationship as a festival has many festival events that are a part of it but a festival event is a part of only one festival. Both Festival\_Event and Festival show total participation for this relationship. This is because a festival must have at least one festival event and a festival event must be part of a festival. This relationship has no independent attributes.
6. **Vendor** sets up booth for **Festival** - This is a many-to-many relationship as one vendor can set up many booths for a festival and one festival can have many vendors setting up booths for it. Both Vendor and Festival show total participation in this relationship. This is because a Vendor must set up at least one booth for a festival and a Festival must have at least one Vendor setting up a booth for it. The independent attributes of this relationship are 4 simple attributes - *Booth\_id, Booth\_location, Contact\_No, POC*. The *Booth\_id* attribute is the primary key for this relationship.

**Conversion to 1NF Relational Schema**



Functional Dependencies –

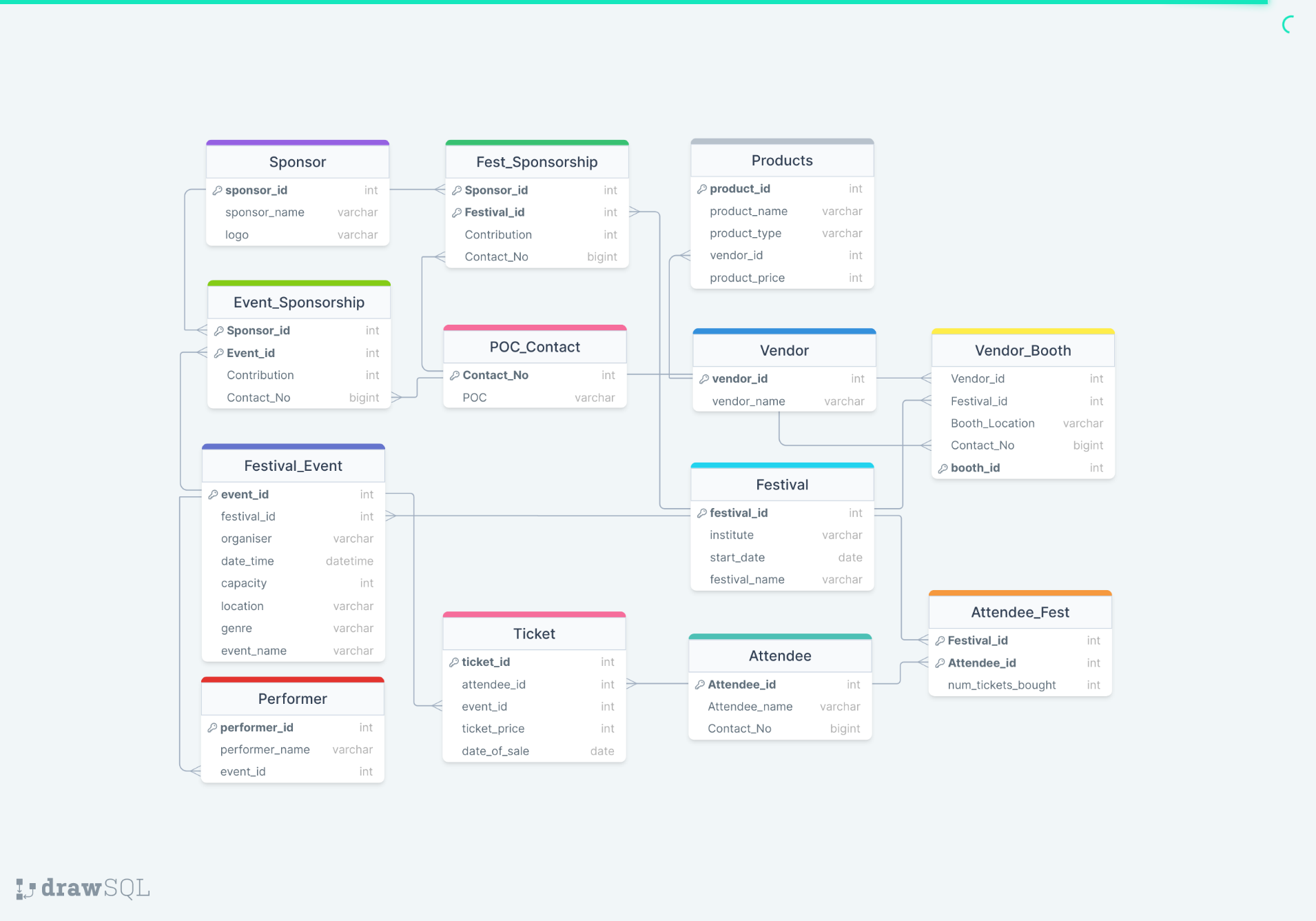
1. **Festival\_Event**  
   Event\_ID 🡪 Festival\_ID, Organiser, Date\_Time, Capacity, Location, Genre,Event\_Name  
   Event\_Name, Festival\_ID 🡪 Event\_ID
2. **Attendee**  
   Attendee\_ID 🡪 Attendee\_name, Contact\_No
3. **Festival**  
   Festival\_ID 🡪 Institute, Start\_Date, Festival\_name
4. **Vendor**  
   Vendor\_ID 🡪 Vendor\_Name
5. **Sponsor**  
   Sponsor\_ID 🡪 Sponsor\_name, Logo

Logo 🡪 Sponsor\_name, Sponsor\_ID

1. **Ticket**  
   Ticket\_ID 🡪 Event\_ID, Attendee\_ID, Ticket\_Price, Date\_of\_Sale
2. **Event\_Sponsorship**Event\_ID, Sponsor\_ID 🡪 Contribution, Contact\_No, POC  
   Contact\_No 🡪 POC, Sponsor\_ID
3. **Fest\_Sponsorship**Festival\_ID, Sponsor\_ID 🡪 Contribution, Contact\_No, POC  
   Contact\_No 🡪 POC, Sponsor\_ID
4. **Vendor\_Booth**Booth \_ID 🡪 Vendor\_ID, Festival\_ID, Booth\_Location, Contact\_No, POC  
   Vendor\_ID, Festival\_ID, Booth\_Location 🡪 Booth\_ID  
   Contact\_No 🡪 POC, Vendor\_ID
5. **Attendee\_Fest**Festival\_ID, Attendee\_ID 🡪 Num\_tickets\_bought
6. **Product**Product\_ID 🡪 Product\_Name, Product\_type, Product\_price, Vendor\_id  
   Vendor\_ID, Product\_Name 🡪 Product\_ID
7. **Performer**Performer\_ID 🡪 Performer\_Name, Event\_ID

The 1st Normal Form does not allow any multi-valued/composite attributes. Therefore, the 2 attributes ‘product\_info’ and ‘performer\_info’ get separated into new tables. All many to many relationships also get created into new tables with their specific attributes. All other relationships remain the same.

**Normalization to 2NF/3NF Relational Schema**



A new table ‘POC\_Contact’ is created. Functional Dependencies of this table are –

Contact\_No 🡪 POC

This was removed as the above functional dependency was a partial dependency in the tables –

* Event\_Sponsorship
* Fest\_Sponsorship
* Vendor\_Booth

This is now stored as a separate table which ensures that all of our tables are now in the 2nd Normal Form.

On observation, we found that we did not have any transitive dependencies in our tables. Therefore, all of our tables are in the 3rd Normal Form.

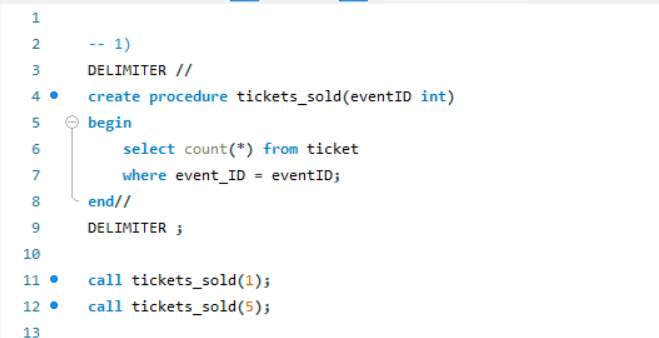
Database

For our database, we have created a total of 13 tables listed below-

1. Festival
2. Festival\_Event
3. Attendee
4. Sponsor
5. Vendor
6. Event\_Sponsorship
7. Fest\_Sponsorship
8. Vendor\_booth
9. Ticket
10. Attendee\_Fest
11. Product
12. Performer
13. POC\_Contact

SQL Queries

Q.1)



Output:

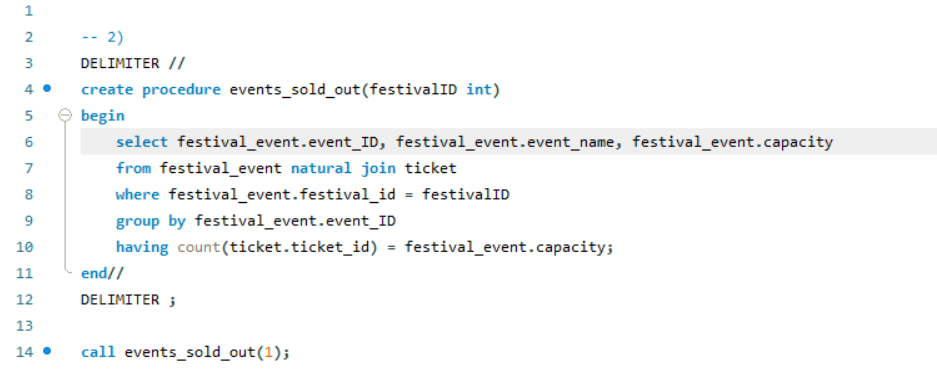
For eventID = 1



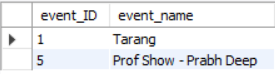
For eventID = 5



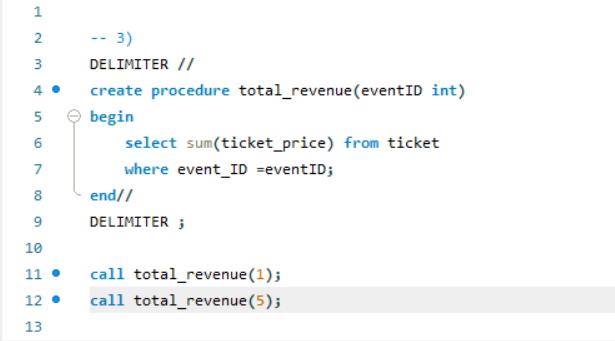
Q.2)



Output:



Q.3)



Output:

For eventID = 1



For eventID = 5



Q.4)



Output:

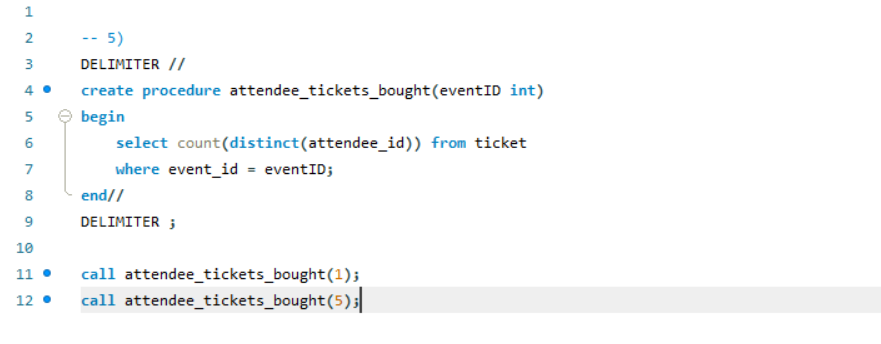
For festivalID = 1



For festivalID = 3



Q.5)



Output:

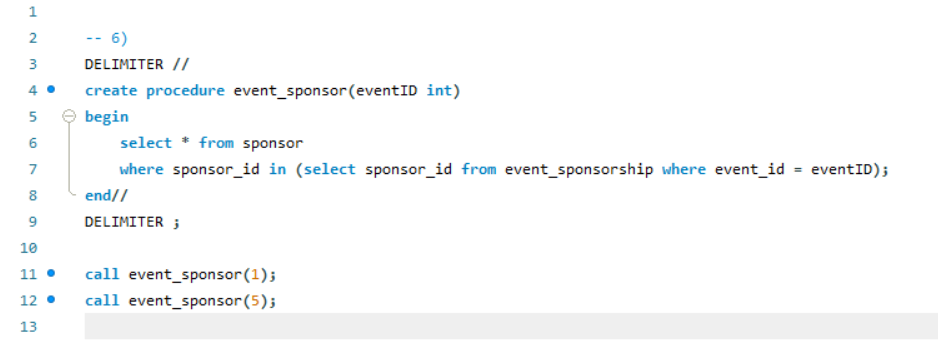
For eventID = 1



For eventID = 5

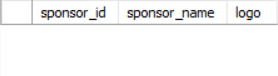


Q.6)



Output:

For eventID = 1

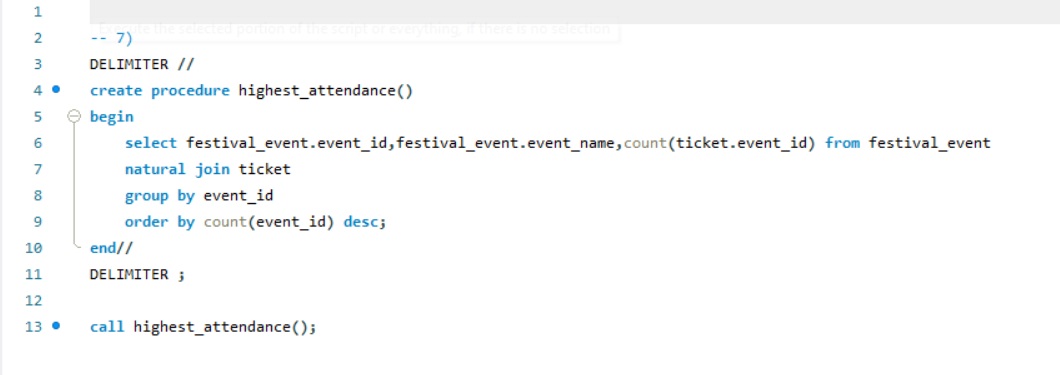


No output 🡪 No Sponsor

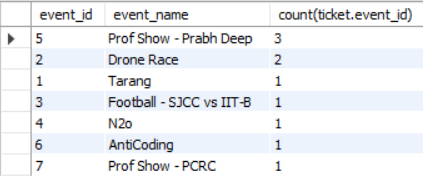
For eventID = 5



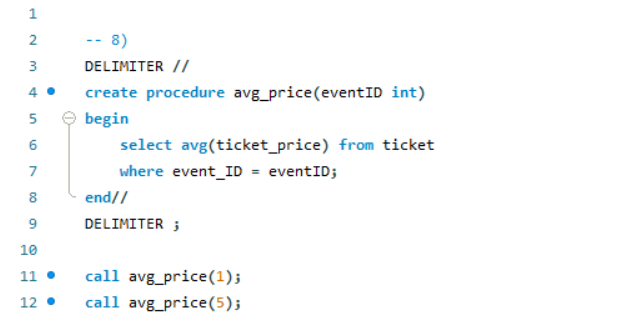
Q.7)



Output:



Q.8)



Output:

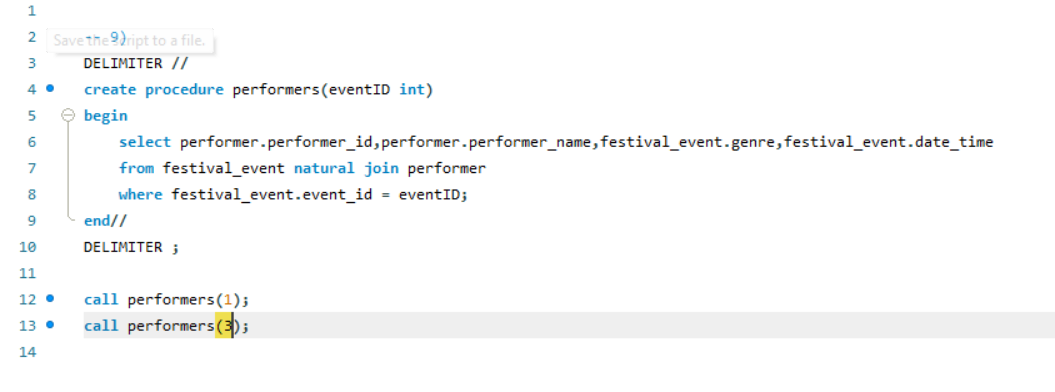
For eventID = 1



For eventID = 5

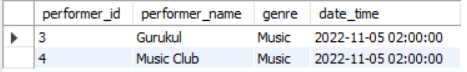


Q.9)



Output:

For eventID = 1

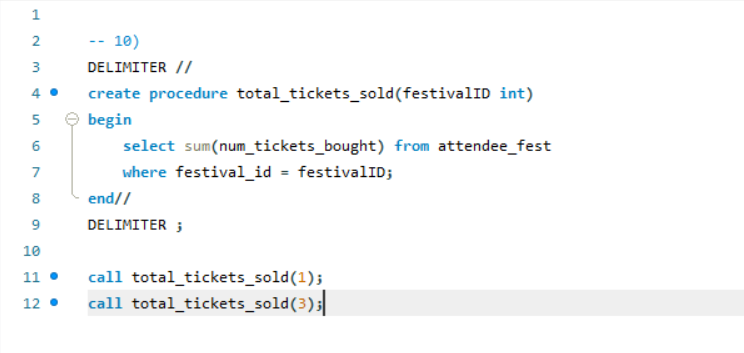


For eventID = 3



No Output 🡪 No Performers

Q.10)



Output:

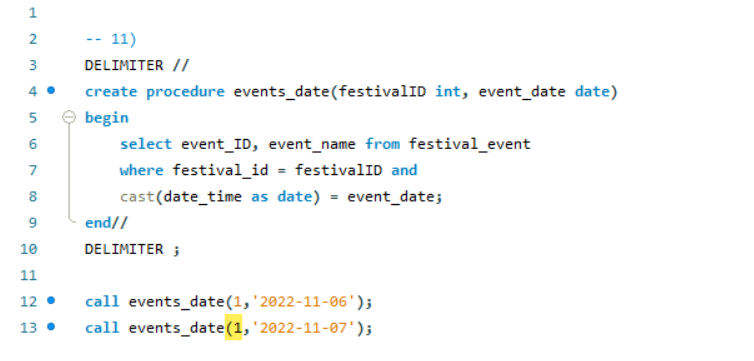
For festivalID = 1



For festivalID = 3



Q.11)

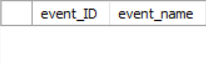


Output:

For event\_date = ‘2022-11-06’

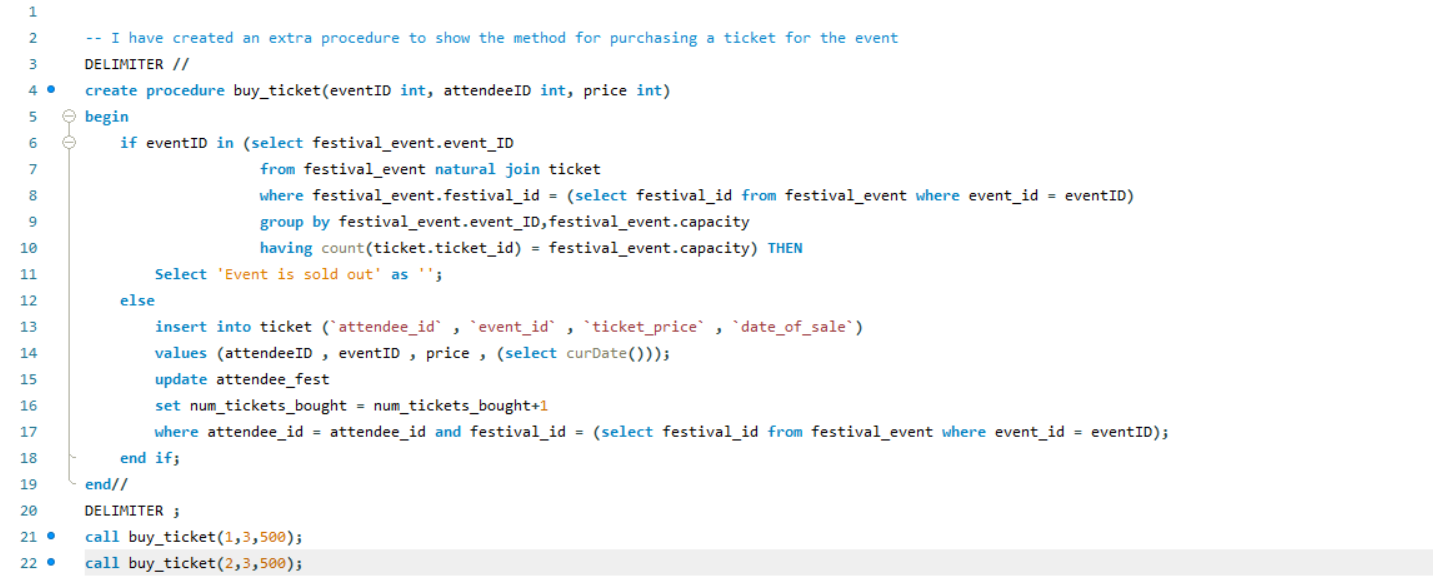


For event\_date = ‘2022-11-07’



No Output 🡪 No Performers

**Procedure to buy ticket**

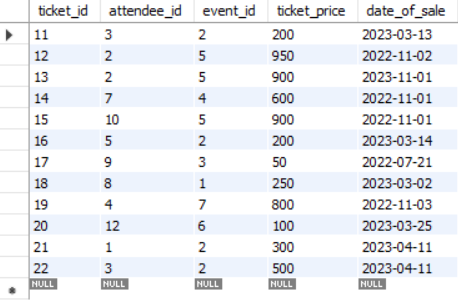


Output:

For eventID = 1



For eventID = 2

  
The last record was added upon the execution of the procedure.

Drive Link for videos-

<https://drive.google.com/drive/folders/1keU4HZHsEJHhDx0C79aSintA8MiV3_V_>